



S&amp;H Form: (02/05)

AF  
JR**REPLY BRIEF  
FEE TRANSMITTAL**

Attorney Docket No.	1046.1236
Application Number	09/778,076
Filing Date	February 7, 2001
First Named Inventor	Yutaka HAGA
Group Art Unit	2192

AMOUNT ENCLOSED

0.00

Examiner Name

Yigdall, Michael J.

**FEE CALCULATION (fees effective 12/08/04)**

CLAIMS AS AMENDED	Claims Remaining After Amendment	Highest Number Previously Paid For	Number Extra	Rate	Calculations
TOTAL CLAIMS	33	- 39 =	0	X \$ 50.00 =	\$ 0.00
INDEPENDENT CLAIMS	3	- 3 =	0	X \$ 200.00 =	0.00

Since an Official Action set an original due date of December 24, 2006, petition is hereby made for an extension to cover the date this reply is filed for which the requisite fee is enclosed (1 month (\$120)); (2 months (\$450)); (3 months (\$1,020)); (4 months (\$1,590)); (5 months (\$2,160)):

If Appeal Brief is enclosed, add (\$500.00)

If Statutory Disclaimer under Rule 20(d) is enclosed, add fee (\$130.00)

Information Disclosure Statement (Rule 1.17(p)) (\$180.00)

Total of above Calculations =

\$ 0.00

Reduction by 50% for filing by small entity (37 CFR 1.9, 1.27 & 1.28)

TOTAL FEES DUE =

\$ 0.00

- (1) If entry (1) is less than entry (2), entry (3) is "0".  
(2) If entry (2) is less than 20, change entry (2) to "20".  
(4) If entry (4) is less than entry (5), entry (6) is "0".  
(5) If entry (5) is less than 3, change entry (5) to "3".

**METHOD OF PAYMENT**

- ☐ Check enclosed as payment.  
☐ Charge "TOTAL FEES DUE" to the Deposit Account No. below.  
☐ No payment is enclosed.

**GENERAL AUTHORIZATION**

- ☒ If the above-noted "AMOUNT ENCLOSED" is not correct, the Commissioner is hereby authorized to credit any overpayment or charge any additional fees necessary to:
- |                      |                    |
|----------------------|--------------------|
| Deposit Account No.  | 19-3935            |
| Deposit Account Name | STAAS & HALSEY LLP |
- ☒ The Commissioner is also authorized to credit any overpayments or charge any additional fees required under 37 CFR 1.16 (filing fees) or 37 CFR 1.17 (processing fees) during the prosecution of this application, including any related application(s) claiming benefit hereof pursuant to 35 USC § 120 (e.g., continuations/divisionals/CIPs under 37 CFR 1.53(b) and/or continuations/divisionals/CPAs under 37 CFR 1.53(d)) to maintain pendency hereof or of any such related application.

**SUBMITTED BY: STAAS & HALSEY LLP**

Typed Name Paul W. Bobowiec

Reg. No.

47,431

Signature

Date

August 7, 2007

©2005 Staas &amp; Halsey LLP



Docket No. 1046.1236

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re Patent Application of:

Yutaka HAGA

Application No.: 09/778,076

Group Art Unit: 2192

Filed: February 7, 2001

Examiner: Yigdall, Michael J.

For: APPARATUS FOR COLLECTING PROFILES OF PROGRAMS

**REPLY BRIEF UNDER 37 CFR §41.41**

Commissioner for Patents  
PO Box 1450  
Alexandria, VA 22313-1450

Sir:

Entry of this Reply Brief is respectfully requested. This Reply Brief is submitted in response to the Examiner's Answer mailed June 18, 2007 ("Examiner's Answer") in reply to the Substitute Appeal Brief filed February 20, 2007 ("Appeal Brief").

**I. STATUS OF CLAIMS**

Claims 8-17, 19-28, and 30-42 are pending in this application at the filing of this Reply Brief. Claims 8-17, 19-28, and 30-42 have at least been twice rejected. Claims 12, 23, and 34 are independent claims, and claims 8-11, 13-17, 19-22, 24-28, 30-33, and 35-42 are dependent claims.

**II. GROUNDS OF REJECTION**

Claims 8-17, 19-28, and 30-42 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Alexander, III et al. (U.S.P. 6,002,872) (Alexander) in view of Smolders (U.S.P. 6,253,338) (Smolders) and Yeh et al. (U.S.P. 6,427,206) (Yeh).

**III. REPLY ARGUMENT**

Beginning on page 13 of the Examiner's Answer, the Examiner is rebutting Appellant's argument from the Appeal Brief that recited features in claim 12 of a method "identifying a type of said branch instruction by obtaining an instruction code from said branch source address and decoding said instruction code; and . . . and when the identified branch instruction is neither a calling instruction nor a return instruction, said interrupt is terminated" are not discussed by an

*arguendo* combination of Alexander in view of Smolders and in view of Yeh. Further, the Examiner is rebutting Appellant's argument from the Appeal Brief that the Examiner had not provided required support for the statement that it is obvious to "substitute the timer interrupt of Alexander with the branch interrupt of Smolders" and that the statement is, rather, a conclusory statement that does not properly support an establishment of *prima facie* obviousness.

In item (9) of the Examiner's Answer, entitled Grounds of Rejection, the Examiner asserts that Alexander discloses:

an apparatus for collecting a profile of a subroutine included in a program (see, for example, the title and abstract), comprising: . . .

(b) an analyzing section, when an interrupt is generated during execution of said program: obtaining a branch source address and a branch destination address from a source of said interrupt (see, for example, column 5, lines 20-32, which shows analyzing the stack frames in response to an interrupt to identify subroutines, and column 5, lines 41-62, which shows obtaining a call or branch source address and a return or branch destination address).

(Examiner's Answer at pages 3-4).

In an Amendment filed July 30, 2004, Appellants argued that there was no motivation to modify Alexander to replace timer interrupt with a branch interrupt and that "a combination of the art in itself would require code modification."

However, the Examiner continues to assert it would have been obvious to one of ordinary skill in the art:

to substitute the timer interrupt of Alexander with the branch interrupt of Smolders, as suggested by Alexander, so as to obviate any overhead and modifications to the code.

(Examiner's Answer at page 5).

The Examiner asserts:

Smolders' teachings enable "tracing hardware counters by way of an interruption without introducing any overhead or modifying the code" (see, for example, column 1, lines 64-67). Therefore, it would have been obvious to one of ordinary skill in the art to use a branch interrupt to trigger the sampling mechanism of Alexander. As Smolders suggests, this would enable sampling without introducing any overhead and without modifying the code of the program.

(Examiner's Answer at page 14).

In reply, Applicants submit that:

1) Being merely able to *arguendo* substitute a timer interrupt with a branch interrupt, even if no code was required to be modified, is not, in itself, a motivation for such a substitution. Using the Examiner's logic, a motivation exists for modification of anything that could possibly be modified, which clearly does not support an establishment of *prima facie*

obviousness; and

2) The Examiner misapplies, or misinterprets, Smolder's teaching regarding modification of code as somehow implying that substituting a branch interrupt for a timer interrupt does not introduce any overhead nor modify the code of the program.

However, Smolders clearly qualifies the teaching as:

Application Programming Interfaces (API) have also been built to collect counter information for portions of workloads. In this mode, one must add calls to API code just before and immediately after the execution of the portion of the workload to analyze. The former is to setup and start the counting and the latter is to stop the counting and retrieve the results. Those API calls can either be added directly to the source code if it is available or by way of instrumentation, i.e. dynamic insertion of code to divert normal execution path. . . . Consequently, it would be desirable to provide an improved method and system that determines which part of a workload is responsible for counter increments of desired events without any overhead. . . . It is yet another object of the present invention to provide an improved method and system for tracing hardware counters by way of an interruption without introducing any overhead or modifying the code.

(Emphasis added, col. 1, lines 33-68).

After Smolders discusses in columns 2-5 the coding and features of his invention for tracing hardware counters, Smolder's concludes:

the advantage of using the present invention versus prior art global counting tools and application programming interfaces is that the counter level tracing tool does not introduce any overhead and therefore allows fine analysis of the counter increments at the basic block level. When using prior art techniques, global counting tools, no overhead is introduced either, but the granularity of the results is from start to finish, making fine analysis impossible. When using prior art API, lower granularity of results can be produced but only up to a certain limit determined by the overhead introduced. The overhead is caused by the counting of the API code itself. With the present invention, the code of the counter level tracing tool is not counted therefore introducing no overhead at all.

(emphasis added, col. 5, line 65 - col. 6, line 14).

That is, Smolder's merely teaches a way of tracing hardware counters without adding code that is counted as API code. The counter level tracing tool is not a codeless modification thought of a programmer, but rather is code/modifications as described in cols 2-5 of Smolder. However, the counter level tracing tool code, itself, is not counted, i.e., no overhead, and does not modify the API code.

However, Smolder does not teach that a timer interrupt can be introduced into a program without any writing of code at all, as the Examiner seems to imply and looks to as a motivation.

In rebutting Appellants argument that since Alexander clearly distinguishes, for one of

ordinary skill in the art a difference between different types of interrupts at there is no motivation to modify Alexander with Smolders, the Examiner asserts:

the page fault interrupt of Alexander is a recurring event in the same sense that the trace interrupt of Smolders is a recurring event. While another interpretation of a "periodic event" is an event that occurs at a fixed frequency or interval, one of ordinary skill in the art would recognize that the page fault interrupts of Alexander are not necessarily generated at a fixed frequency or interval. Thus, a conclusion that the sampling mechanism of Alexander is strictly limited to interrupts that are fixed-frequency events is not supported. The discussion of the page fault interrupt in Alexander is, at minimum, evidence in-favor of a reasonable expectation of success in substituting the timer interrupt with another recurring event such as Smolders' trace interrupt.

(Examiner's Answer at page 15).

Again, the Examiner is attempting to establish motivation by merely arguing a modification is *arguendo* possible. However, Appellants in reply point out that the Alexander clearly distinguishes between the utility and function of the timer fault and the page fault. In particular, Alexander discusses

The depicted example illustrated in FIGS. 2 through 10 illustrated a process for structured profiling of data processing systems and applications executing on a data processing system. The depicted example in these figures is directed towards obtaining information from a stack in response to an occurrence of a timer interrupt.

(col. 8, lines 40-50).

Alexander further teaches functional differences between the interrupts and different embodiments of Alexander's invention as:

The processes of the present invention also may be applied to other types of events occurring within a data processing system or application. For example, in another embodiment of the present invention, the processes described may be applied to memory analysis. Instead of employing a timer interrupt, a page fault interrupt may be used as a signal to gather data from the stack. In the depicted example, a page fault is forced each time a memory access occurs within the data processing system. A page fault also may be selectively triggered, depending on processing resources. . . . From this information, the processes of the present invention as described above and additional processes as described below may be employed to analyze memory usage.

(col. 8, starting at line 48)

Appellant points out that claim 12 recites an apparatus directed to collecting a profile of a subroutine included in a program, and that it is not obvious in view of Alexander that interrupts can be substituted for "collecting a profile of a subroutine." Appellant submits that Alexander teaches different interrupts may be used for different purposes, but does not teach a blanket substitution of the timer interrupt with a page fault, as the Examiner appears to imply. Thus, the

assertion by the Examiner that interrupts can be substituted is an unsupported conclusory statement and is further contradicted by the art, itself, relied on by the Examiner.

The Examiner further asserts it would have been obvious

to supplement the profiling system of Alexander with the features taught by Yeh and to identify the type of branch, as taught by Yeh, so as to collect branch predictions and enhance the collection of profiles for purposes of speculative execution.

(Examiner's Answer at page 5).

Claim 12 recites "obtaining a branch source address and a branch destination address from a source of said interrupt." Appellant points out that if Alexander was modified to include a page fault than other features of claim 12 would not be taught by the cited art, since a source of the interrupt would not be a branch source address and a branch destination address but something else.

Further, Appellant points out that claim 12 recites " when the identified branch instruction is neither a calling instruction nor a return instruction, said interrupt is terminated (emphasis added)." The Examiner asserts

when the identified branch instruction is neither a calling instruction nor a return instruction, said interrupt is terminated (see, for example, Yeh, step 310 in FIG. 3 and column 8, lines 42-48, which shows not collecting history information when the branch is not a calling instruction or a return instruction, and see, for example, Smolders, steps 46 and 64 in FIG. 3 and column 5, lines 43-52, which shows terminating the interrupt when trace information is not to be collected).

(Examiner's Answer at page 6).

Again, the Examiner is attempting to support establish motivation by merely asserting a modification is *arguendo* possible.

Appellant submits that similar replies can be made in behalf of independent claims 23 and 34.

Appellant respectfully points out that points out in *KSR International Co. v. Teleflex Inc.*, 82 USPQ2d 1385, 127 SCt 1727, 167 LE2d 705 (U.S. 2007), the U.S. Supreme Court held that in determining obviousness, one "must ask whether the improvement is more than the predictable use of prior art elements according to their established functions" slip op. 13, 82 USPQ2d at 1396. Furthermore, it is necessary "to determine whether there was an apparent reason to combine the known elements in the fashion claimed" slip op. 14, 82 USPQ2d at 1396.

The Supreme Court further affirmed *In re Kahn*, 441 F. 3d 977, 988 (CA Fed. 2006), stating: "[R]ejections on obviousness grounds cannot be sustained by mere conclusory

statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." As relied upon in the *KSR* decision, any underlying obvious to try rationale still requires evidence in the record of the same.

As pointed out in the Appeal Brief, using the Examiner's logic each and every feature of Smolders or Yeh, or any other reference would have been obvious to modify into Alexander if there was a possibility that it could somehow be done, regardless of a desire/need/benefit between this feature.

Appellant submits that the Examiner support are mere conclusory statements, and as stated by the Supreme Court *In re Kahn* "[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements."

Starting on page 17 of the Examiner Answer, the Examiner rebuts Appellant's argument that claim 40 recites an apparatus including specific recited names and labels given to the tables in a certain relationship that render the same "functionally distinct" from an arbitrary association of pointers and nodes referred to by the Examiner by asserting that the claim 40 recited subject matter "amounts to a data structure that includes arbitrarily named tables."

In reply to the Examiner's Answer, Appellant submits that as such data structures not "arbitrarily" named. (See, for example, Merriam- Webster Dictionary that defines the term "arbitrarily" as "based on or determined by individual preference or convenience rather than by necessity or the intrinsic nature of something existing" or as "coming about seemingly at random or by chance or as a capricious and unreasonable act of will at <<http://www.m-w.com/cgi-bin/dictionary?book=Dictionary&va=arbitrarily>>).

Claim 40 recites an apparatus including a" table that is generated corresponding to each executor of the subroutine on the storage unit, . . . executor managing table stores an identifier of the executor and a pointer to assign the subroutine managing table, wherein the "subroutine managing table is generated for every subroutine executed by the executor, the subroutine managing table storing a subroutine address, times of calling of the subroutine, a cumulative execution time of the subroutine, the last called time of the subroutine, and a pointer to assign the calling managing table, and wherein the calling managing table is generated for every subroutine called by the subroutine, the calling managing table storing a branch source address as a calling subroutine address, a branch destination address as a called subroutine address, times of calling of the called subroutine, a cumulative execution time of the called subroutine, the last called time of the called subroutine, and a pointer to specify the subroutine managing table managing the calling subroutine. (emphasis added)."

That is, claim 40 recites features that are functionally interrelated with the names of features indicating, in part, such a functional relationship and actions.

The Examiner asserts:

One of ordinary skill in the art would recognize that such pointers are necessary to allow the elements of the data structure to be located, accessed, traversed and so on. . . . Those of ordinary skill [in the] art will appreciate that tree structure 500 may be implemented in a variety of ways and that many different types of statistics may be maintained at the nodes other than those in the depicted example. In addition, other pointers may be stored within the nodes to further aid subsequent analysis. Further, other structural elements, such as tables for properties of the routine, such as, for example, the name of the routine, also may be stored within a node. . . . it would have been obvious to one of ordinary skill in the art to implement the data structure of Alexander in a form suitable for subsequent analysis. That is to say, Alexander suggests that one of ordinary skill in the art might choose to implement the data structure in any form suitable for the subsequent analysis that he or she intends to perform. Again, as noted above, the information that Alexander's data structure provides is analogous to the information that the claimed data structure provides.

(Examiner's Answer at page 18).

Appellant respectfully submits that using the Examiner's logic that since pointers are "necessary," and trees can be implemented in a variety of ways that any specific recited relationship between one table and another, including generation of a specific is obvious in view of Alexander, regardless of the implementation.

Again, the Examiner is attempting to establish motivation by merely arguing a modification is *arguendo* possible. The Examiner has not provided support for why such a modification of Alexander "to implement the data structure in any form suitable" should be accomplished. Appellant respectfully submits that the Examiner's answers are mere conclusory statements, and as stated by the Supreme Court *In re Kahn* "[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements."

#### IV. CONCLUSION

In view of the law and facts stated herein, the Appellant respectfully submits that the Examiner has failed set forth a *prima facie* obviousness case against the pending claims.

For all the foregoing reasons, the Appellant respectfully submits that the cited prior art does not teach or suggest the presently claimed invention. The claims are patentable over the prior art of record and the Examiner's findings of unpatentability regarding claims 8-17, 19-28 and 30-42 should be reversed and the patentability over the presently cited references be affirmed.



The Commissioner is hereby authorized to charge any additional fees required in connection with the filing of this Reply Brief to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

Date: August 7, 2007

By Paul W. Bobowiec  
Paul W. Bobowiec  
Registration No. 47,431

1201 New York Ave, N.W., Suite 700  
Washington, D.C. 20005  
Telephone: (202) 434-1500  
Facsimile: (202) 434-1501